



Hans Hagen

# Modes

Every user will at one moment run into modes. Modes are used for conditional processing. You enable or disable modes:

```
\enablemode[screen]
\disablemode[proof]
```

as well as prevent modes being set:

```
\preventmode[doublesided]
```

Later on you can act upon this mode using:

```
\startmode[screen]
  \setupinteraction[state=start]
\stopmode
```

The counterpart of this command is:

```
\startnotmode[screen]
  \setupinteraction[state=start]
\stopnotmode
```

You can set modes in your document or in styles, but you can also do that at runtime:

```
texexec --pdf --mode=screen --result=myfile-s myfile
texexec --pdf --mode=A4      --result=myfile-a myfile
texexec --pdf --mode=letter  --result=myfile-l myfile
```

You can test for more modes at the same time:

```
\startmode[color,colour]
  \setupcolors[state=start]
\stopmode
```

If you want to satisfy a combination of modes, you use:

```
\startmode[final]
  \setuplayout[markings=on]
\stopmode
\startallmodes[final,color]
  \setuplayout[markings=color]
\stopallmodes
```

The counterpart is

```
\startnotallmodes[print,proof]
  \setuplayout[markings=off]
\stopnotallmodes
```

Instead of the `start-stop` variants, you can use the `\doif` alternatives. These have the advantage that they can be nested.

```
\doifmodeelse      {modes} {action} {alternative}
\doifmode          {modes} {action}
\doifnotmode       {modes} {action}
\doifallmodeselse {modes} {action} {alternative}
\doifallmodes      {modes} {action}
\doifnotallmodes   {modes} {action}
```

Mode can be combined with variables:

```
\setupvariables[document][alternative=print]

\enablemode[document:\getvariable{document}{alternative}]

\startmode[document:print]
  ...
\stopmode

\startmode[document:screen]
  ...
\stopmode
```

An alternative for such an selective approach is to use setups:

```
\setupvariables[document][alternative=print]

\startsetups[document:print]
  ...
\stopsetups

\startsetups[document:screen]
  ...
\stopsetups

\setups[document:\getvariable{document}{alternative}]
```

The difference is that mode blocks are processed in the order that the document (or style) is loaded, while setups are stored and recalled later.

In addition to your own modes, `CONTEXT` provides a couple of system modes. These are preceded by a `*`, as in:

```
\startmode[*first]
  % this is the first run
\stopmode
```

The following system modes are available (more will be implemented):

`color-c`, `color-m`, `color-y`, `color-k` These are rather special modes related to color separation. They are only set when channels are split off.

`figure` This mode is set when a graphic is found. You can use this mode in for instance figure postprocessing actions.

`text`, `project`, `product`, `component`, `environment` These modes are set when one enters one of the associated structuring environments. Nesting is supported.

`list` After using `\determinelistcharacteristics` this mode reflects if list entries were found.

`pairedbox` This mode is enabled when a paired box (legenda and such) is constructed.

`combination` This mode is enabled when a combination (often used for graphics) is constructed.

`interaction` When interaction is enabled, this mode is true. You can for instance use this mode to add different content to for instance screen and paper versions.

`register` After using `\determineregistercharacteristics` this mode reflects if register entries were found.

`sectionnumber` This mode is enabled when a section head is numbered. You can access the mode while building the section head, which is true when you have your own commands hooked into the head mechanism.

`frontpart`, `bodypart`, `backpart`, `appendix` The state of main sections in a document as well as user defined ones, are reflected in system modes.

**suffix-tex** You can use this mode to differentiate between input file types. We use this for instance to distinguish between different XML content variants when pretty-printing (given that they can be recognized on their suffix).

**first** Often multiple runs are needed to get a document right. Think of cross references, object references, tables of contents, indices, etc. You can use this mode to determine if the first run is taking place. For instance, when you do real time graphic conversions, it makes sense to do that only once.

**last** This mode is set if the last run in a session is taking place. Normally this is not known in advance, unless one has asked for an additional imposition pass.

**background** This mode is set when there is a (new) background defined.

**postponing** While postponing some content using the postpone mechanism this mode is enabled.

**grid** When you are typesetting on a grid, special care has to be taken not spoil grid snapping. You can use this mode to test if you are in grid typesetting mode.

**header** This mode is enabled when there is a page header, i.e. the header has non-zero dimensions.

**footer** This mode is enabled when there is a page footer, i.e. the header has non-zero dimensions.

**makeup** The makeup mechanisms are used to build single pages like title pages. This mode is set during construction.

**pdf, dvi** One of these modes is set, which one depends on the output driver that is loaded.

**\*language-id, language-id** When a language is chosen, its id is set as mode. For example, when the main language is English, and the current language Dutch, we can test for the modes **\*\*en** and **\*nl** (watch the extra **\***).

**\*marking** This flag is set when a marking (e.g. in a header or footer) is being typeset (processed).